Incorporating Term Dependency in the DFR Framework

Jie Peng*, Craig Macdonald*, Ben He*, Vassilis Plachouras†, Iadh Ounis*

*University of Glasgow
Glasgow, UK
{pj, craigm, ben, ounis}@dcs.gla.ac.uk

†Yahoo! Research
Barcelona, Spain
vassilis@yahoo-inc.com

ABSTRACT

Term dependency, or co-occurrence, has been studied in language modelling, for instance by Metzler & Croft [3] who showed that retrieval performance could be significantly enhanced using term dependency information. In this work, we show how term dependency can be modelled within the Divergence From Randomness (DFR) framework. We evaluate our term dependency model on the two adhoc retrieval tasks using the TREC .GOV2 Terabyte collection. Furthermore, we examine the effect of varying the term dependency window size on the retrieval performance of the proposed model. Our experiments show that term dependency can indeed be successfully incorporated within the DFR framework.

Categories and Subject Descriptors: H.3.3 [Information Storage & Retrieval]: Information Search & Retrieval

General Terms: Performance, Experimentation

Keywords: Term Dependency, DFR

1. INTRODUCTION

Document weighting models, such as BM25 and language modelling, rank documents using the occurrences of single query terms in documents and assuming that the query terms are independent. However, previous studies have shown that taking the dependency of query terms in documents into account can improve retrieval performance [3, 4]. In particular, Metzler & Croft have developed a formal framework for modelling term dependencies via Markov random fields [3]. They explored three possible relation variants between query terms: Full Independence, which assumes query terms are independent with each other; Sequential Dependence, which only assumes a dependence between neighbouring query terms; and Full Dependence which assumes all query terms are dependent with each other. Their experiments showed that their term dependency model could significantly improve retrieval performance.

In this paper, we show how term dependency can be naturally modelled within the Divergence From Randomness framework [1] (Section 2). We evaluate the proposed model in the context of the TREC 2005 and TREC 2006 Terabyte track adhoc tasks (Section 3). In particular, we observe comparable conclusions to Metzler & Croft's [3], namely that the incorporation of term dependencies into a DFR-based model can significantly enhance retrieval performance.

SIGIR' 07, July 23–27, 2007, Amsterdam, The Netherlands.
ACM 978-1-59593-597-7/07/0007.

2. TERM DEPENDENCE IN THE DFR FRAMEWORK

We use the DFR paradigm to capture the dependence of query terms in documents. The proposed model assigns scores to pairs of query terms, in addition to the single query terms. Hence the score of a document d for a query Q is given as follows:

$$score(d, Q) = \lambda_1 \cdot \sum_{t \in Q} score(d, t) + \lambda_2 \cdot \sum_{p \in Q_2} score(d, p) \quad (1)$$

where $score(d, t)$ is the score assigned to a query term t in the document d, p corresponds to a pair of query terms that appear within the query Q, $score(d, p)$ is the score assigned to a pair of query terms p in the document d, and Q_2 is a set of pairs of query terms, as defined below. The two scores are combined linearly using λ_1 & λ_2 as weights. For simplicity, we use only binary weights. In Equation (1), the score $\sum_{t \in Q} score(d, t)$ can be estimated by any DFR weighting model. For example, we can use the DFR PL2 document weighting model [1].

We now introduce three possible variants in using term dependencies between query terms: For full independence (FI), the introduced weighting model only computes the first component of Equation (1) as it ignores the term dependencies between query terms ($\lambda_1 = 1$, $\lambda_2 = 0$). This is equivalent to PL2 alone; For sequential dependence (SD), we compute both components of Equation (1) ($\lambda_1 = 1$, $\lambda_2 = 1$), and in this case, Q_2 is the set that contains *ordered* pairs of neighbouring query terms; For full dependence (FD), we compute both components of Equation (1) ($\lambda_1 = 1$, $\lambda_2 = 1$), and in this case, Q_2 is the set that contains *unordered* pairs of any two query terms. In this paper, we consider only pairs of query terms, even if we could easily extend the model to more than two terms. The weight $score(d, p)$ of a pair of query terms in a document is computed as follows:

$$score(d, p) = -\log_2(P_{p1}) \cdot (1 - P_{p2}) \quad (2)$$

where P_{p1} corresponds to the probability that there is a document in which a pair of query terms p occurs a given number of times. P_{p1} can be computed with any Randomness model from the DFR framework. P_{p2} corresponds to the probability of seeing the pair of query terms once more, after having seen it a given number of times. P_{p2} can be computed using any of the After-effect models in the DFR framework. The difference between $score(d, p)$ and $score(d, t)$ is that the former depends on counts of occurrences of the pair of query terms p, while the latter depends on counts of occurrences of the query term t.

	TREC 2006 adhoc Task			TREC 2005 adhoc Task		
	MAP	b-Pref	P@10	MAP	b-Pref	P@10
FI	0.3062	0.3572	0.5640	0.3407	0.3770	0.6180
SD	0.3175$*$ (+3.7%)	0.3670$*$ (+2.7%)	**0.5940 (+5.3%)**	0.3384 (-0.2%)	0.3767 (≈0.0%)	**0.6280 (+1.6%)**
FD	**0.3297$**$ (+7.7%)**	**0.3811$**$ (+6.7%)**	0.5600 (-0.7%)	**0.3488 (+2.4%)**	**0.3868 (+2.6%)**	0.6240 (+1.0%)

Table 1: MAP, b-Pref, and P@10 scores for each variant for TREC 2005 and TREC 2006 adhoc task, respectively. Values in parenthesis denote percentage improvement over full independence (FI). Scores statistically improved over FI are marked with $*$ and scores statistically improved over both FI and SD are marked with $**$ (Wilcoxon Matched-Pairs Signed-Ranks Test, $p < 0.05$).The best result in each column is emphasised.

To compute the weight $score(d,p)$, we use a Randomness model, which does not consider the collection frequency of the pair of query terms. It is based on the binomial Randomness model, given as follows [2]:

$$score(d,p) = \frac{1}{pfn+1} \cdot \Big(- \log_2 (l-1)! + \log_2 pfn! + \log_2 (l-1-pfn)! - pfn \log_2(p_p) - (l-1-pfn)\log_2(p'_p)\Big) \quad (3)$$

where l is the length of the document in tokens, $p_p = \frac{1}{l-1}$, $p'_p = 1 - p_p$, and pfn is the normalised frequency of the pair of query terms p using Normalisation 2 [1]:

$$pfn = pf \cdot \log_2(1 + c_p \cdot \frac{avg\_l - 1}{l-1})(c_p > 0) \quad (4)$$

pf is the frequency of the pair of query terms p that appear within $window\_size$ tokens in the document (for SD, these must appear in the same order as in pair p), avg_l is the average document length in the collection, and c_p is a hyper-parameter that controls the normalisation applied to the pair of query terms frequency against document length.

3. EXPERIMENTS AND ANALYSIS

We use the TREC .GOV2 Terabyte test collection, and its associated TREC 2005 and 2006 adhoc title-only topics and relevance assessment sets. For indexing and retrieval we use Terrier[1], with Porter's weak stemming and removing stopwords. For all our experiments, we set $c = 6$ and $c_p = 0.05$ for the term frequency normalisation parameter of PL2 and the pair of query terms frequency normalisation respectively, as suggested by [2]. For the $window\_size$, we use 5 – the default setting suggested by [2]. We report Mean Average Precision (MAP), binary preference (b-Pref), and Precision at 10 (P@10).

Firstly, in Table 1, we assess the extent to which the incorporation of term dependency enhances retrieval performance over a full independence baseline (FI). We observe that the FD variant always outperforms the baseline, except for P@10 on the TREC 2006 queries (the improvements on the TREC 2006 queries are statistically significant). Moreover, the SD variant outperforms the FI baseline (these improvements are statistically significant for MAP and b-Pref on the TREC 2006 queries), except for MAP and b-Pref on the TREC 2005 queries, which are slightly (but not significantly) lower than the baseline.

Secondly, we compare the SD and FD variants. From Table 1, we observe that while FD outperforms SD for MAP and b-Pref (these improvements are significant on the TREC 2006 queries), P@10 is enhanced more by SD than by FD.

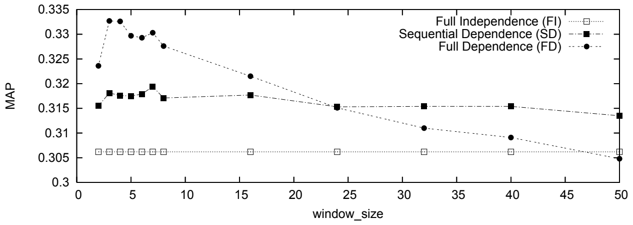

Figure 1: MAP distribution for TREC 2006 for varying $window\_size$.

Moreover, we vary the $window\_size$ parameter to investigate its effect on retrieval effectiveness. For lack of space, we consider only the TREC 2006 queries and the MAP measure. From Figure 1, we observe that both SD and FD, in most cases, outperform the FI baseline. The performance is stable for the SD variant as $window\_size$ is varied. However, the improvement decreases as $window\_size$ is increased for the FD variant. In general, a $window\_size$ around 5 will produce the best MAP scores for the FD variant. Furthermore, for $2 \leq window\_size \leq 24$, the FD variant outperforms SD, while for $window\_size > 24$, SD performs better.

Finally, we notice that the results observed in our experiments mirror those observed in [3], namely that FD can significantly outperform SD, and both FD and SD can significantly outperform the FI baseline, even though, unlike [3], we only consider pairs of query terms. Furthermore, our retrieval performance would likely be enhanced by suitable training of λ_1 & λ_2 in Equation (1).

4. CONCLUSIONS

We have introduced a novel DFR-centric approach for incorporating term dependencies. Our approach is shown to be robust as retrieval effectiveness is enhanced on a large-scale adhoc test collection, using a variety of window sizes. Further enhancement of retrieval performance is likely to be achieved by an appropriate training of the model, similar to that in [3].

5. REFERENCES

[1] G. Amati. *Probability Models for Information Retrieval based on Divergence From Randomness*. PhD thesis, University of Glasgow, 2003.

[2] C. Lioma, C. Macdonald, V. Plachouras, J. Peng, B. He, and I. Ounis. Experiments in Terabyte and Enterprise tracks with Terrier. In *Proceedings of TREC 2006*, 2007.

[3] D. Metzler and W. B. Croft. A Markov Random Field Model for Term Dependencies. In *Proceedings of SIGIR 2005*, pages 472–479. 2005.

[4] M. Srikanth and R. Srihari. Biterm Language Models for Document Retrieval. In *Proceedings of SIGIR 2002*, pages 425–426. 2002.

[1] URL: `http://ir.dcs.gla.ac.uk/terrier/`